

Model Development Phase

Date	13 June 2025
Team ID	SWTID1749627644
Project Title	Human Resource Management: Predicting Employee Promotions using Machine Learning
Maximum Marks	4 Marks

Initial Model Training Code, Model Validation and Evaluation Report:

The initial model training code will be showcased in the future through a screenshot. The model validation and evaluation report will include classification reports, accuracy, and confusion matrices for multiple models, presented through respective screenshots.

Initial Model Training Code:

```
# Decision Tree Model

def decisionTree(x_train, y_train, x_test, y_test):
    dt = DecisionTreeClassifier()
    dt.fit(x_train, y_train)
    y_pred = dt.predict(x_test)
    print('Decision Tree Classifier')
    print('Confusion Matrix:')
    print(confusion_matrix(y_test, y_pred))
    print('Classification Report:')
    print(classification_report(y_test, y_pred))
```

```
# Random Forest Model
```

```
def randomForest(x_train, y_train, x_test, y_test):  
    rf = RandomForestClassifier()  
    rf.fit(x_train, y_train)  
    y_pred = rf.predict(x_test)  
    print('Random Forest Classifier')  
    print('Confusion Matrix:')  
    print(confusion_matrix(y_test, y_pred))  
    print('Classification Report:')  
    print(classification_report(y_test, y_pred))
```

```
# KNN Model
```

```
def KNN(x_train, y_train, x_test, y_test):  
    knn = KNeighborsClassifier()  
    knn.fit(x_train, y_train)  
    y_pred = knn.predict(x_test)  
    print('KNN Classifier')  
    print('Confusion Matrix:')  
    print(confusion_matrix(y_test, y_pred))  
    print('Classification Report:')  
    print(classification_report(y_test, y_pred))
```

```
# XGBoost Model
```

```
def xgboost(x_train, y_train, x_test, y_test):  
    xg = GradientBoostingClassifier()  
    xg.fit(x_train, y_train)  
    y_pred = xg.predict(x_test)  
    print('XGBoost Classifier')  
    print('Confusion Matrix:')  
    print(confusion_matrix(y_test, y_pred))  
    print('Classification Report:')  
    print(classification_report(y_test, y_pred))
```

```
def compareModel(x_train, y_train, x_test, y_test):
    decisionTree(x_train, y_train, x_test, y_test)
    print('- '*50)
    randomForest(x_train, y_train, x_test, y_test)
    print('- '*50)
    KNN(x_train, y_train, x_test, y_test)
    print('- '*50)
    xgboost(x_train, y_train, x_test, y_test)
```

```
compareModel(x_train, y_train, x_test, y_test)
```

Model Validation and Evaluation Report:

Model	Classification Report	F1 Score	Confusion Matrix
Decision Tree Classifier	<pre>Classification Report: precision recall f1-score support 0 0.94 0.92 0.93 15065 1 0.92 0.94 0.93 15019 accuracy 0.93 0.93 0.93 30084 macro avg 0.93 0.93 0.93 30084 weighted avg 0.93 0.93 0.93 30084</pre>	0.93	<pre>Confusion Matrix: [[13881 1184] [918 14101]]</pre>
Random Forest Classifier	<pre>Classification Report: precision recall f1-score support 0 0.95 0.94 0.95 15065 1 0.94 0.95 0.95 15019 accuracy 0.95 0.95 0.95 30084 macro avg 0.95 0.95 0.95 30084 weighted avg 0.95 0.95 0.95 30084</pre>	0.95	<pre>Confusion Matrix: [[14206 859] [781 14238]]</pre>

KNN Classifier	<pre> Classification Report: precision recall f1-score support 0 0.95 0.81 0.88 15065 1 0.84 0.96 0.90 15019 accuracy 0.89 30084 macro avg 0.90 0.89 0.89 30084 weighted avg 0.90 0.89 0.89 30084 </pre>	0.89	<pre> Confusion Matrix: [[12265 2800] [581 14438]] </pre>
XGBoost Classifier	<pre> Classification Report: precision recall f1-score support 0 0.88 0.85 0.86 15065 1 0.85 0.89 0.87 15019 accuracy 0.87 30084 macro avg 0.87 0.87 0.87 30084 weighted avg 0.87 0.87 0.87 30084 </pre>	0.87	<pre> Confusion Matrix: [[12732 2333] [1670 13349]] </pre>